

Chemometrics and Intelligent Laboratory Systems, 20 (1993) 45–55
Elsevier Science Publishers B.V., Amsterdam

Using genetic algorithms for an artificial neural network model inversion

A.P. de Weijer¹, C.B. Lucasius, L. Buydens and G. Kateman

*Department of Analytical Chemistry, Catholic University of Nijmegen, Toernooiveld 1,
6525 ED Nijmegen (Netherlands)*

H.M. Heuvel

Akzo Research Laboratories Arnhem, P.O. Box 9300, 6800 SB Arnhem (Netherlands)

(Received 1 July 1992; accepted 15 January 1993)

Abstract

De Weijer, A.P., Lucasius, C.B., Buydens, L., Kateman, G. and Heuvel, H.M., 1993. Using genetic algorithms for an artificial neural network model inversion. *Chemometrics and Intelligent Laboratory Systems*, 20: 45–55.

Genetic algorithms (GAs) and artificial neural networks (ANNs) are techniques for optimization and learning, respectively, which both have been adopted from nature. Their main advantage over traditional techniques is the relatively better performance when applied to complex relations. GAs and ANNs are both self-learning systems, i.e., they do not require any background knowledge from the creator. In this paper, we describe the performance of a GA that finds hypothetical physical structures of poly(ethylene terephthalate) (PET) yarns corresponding to a certain combination of mechanical and shrinkage properties. This GA uses a validated ANN that has been trained for the complex relation between structure and properties of PET. This technique was tested by comparing the optimal points found by the GA with known experimental data under a variety of multi-criteria conditions.

INTRODUCTION

Artificial neural networks (ANNs) are gaining increasing popularity nowadays. Since the breakthrough in 1986, when a learning rule for com-

plex ANNs was discovered [1], applications have frequently been published in the literature [2–5]. In a previous article, we presented an ANN as a soft modelling technique in the relation between physical yarn structure and properties of poly(ethylene terephthalate) (PET) yarns [6]. It turned out that a set of mechanical and shrinkage properties can be predicted from a set of structure measurements. Our work resulted in an improved understanding of this complex relation [7]. From an effective process and product develop-

Correspondence to: L. Buydens, Department of Analytical Chemistry, Catholic University of Nijmegen, Toernooiveld 1, 6525 ED Nijmegen (Netherlands).

¹ Permanent address: Akzo Research Laboratories Arnhem, P.O. Box 9300, 6800 SB Arnhem (Netherlands).

ment point of view, it is interesting to be able to apply this relation in the reverse way, i.e. to find possible theoretical structures corresponding to a certain combination of desired properties. The above-mentioned approach can be powerful to guide research efforts towards new or optimal production processes.

Since artificial neural network modelling is generally applied to nonlinear and high-dimensional relations, the multi-criteria approach in the reverse relation may contain many (local) optimal points. Direct search algorithms, i.e., methods that seek local optima by moving in a direction related to the local gradient, will miss the main event, when starting in the neighbourhood of local optimal points. This problem belongs to the large class of nondeterministic polynomial time complete (NP-complete) problems since the number of possible solutions and the complexity, i.e. the shape, of the search domain is too high to explore with random or enumerative schemes [8]. No exact solutions of NP-complete problems can directly be calculated, so a heuristic method has been developed to calculate near-optimal solutions. In this paper we will demonstrate that the use of genetic algorithms in a synergic relation with artificial neural networks (SYNGA) is a powerful tool for achieving this. Applications of combinations of ANNs and GAs are described in the literature, but mainly concern optimal design and training of ANNs [9–11].

GENETIC ALGORITHMS

Genetic algorithms (GAs) comprise a family of adaptive search procedures and have been described and analyzed in the literature [12–14]. Their name loosely refers to models based on genetic changes in a population of state vectors s , which may be viewed as generalized chromosomes. Each state vector is composed of a set of one or more parameters s_1, \dots, s_n or generalized genes (Fig. 1).

The genes can be coded in two ways on the state vector. Holland [15] proposed a binary representation on the chromosomes. This representation has successfully been used in several appli-

$$\text{population: } \begin{bmatrix} s_{1,1} & s_{1,2} & \dots & \dots & s_{1,j} \\ s_{2,1} & s_{2,2} & \dots & \dots & s_{2,j} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ s_{i,1} & s_{i,2} & \dots & \dots & s_{i,j} \end{bmatrix}$$

$$\vec{s} = (s_{n,1} \ s_{n,2} \ \dots \ s_{n,j})$$

Fig. 1. A population consisting of i state vectors or generalized chromosomes. Each state vector consists of j parameters or genes.

cations but is not essential to achieve successful evolutions. When a real-coded representation is used, a different parameter setting of the genetic algorithm is required. This point will be discussed later.

Let us also consider a scalar function $U(s)$, the evaluation function which is defined below. This function provides a response for each state s . The subset of states s , enjoying the highest response $U(s)$, represents the solution of the given problem.

The key feature of GAs is their ability to exploit accumulation information about an initially unknown search space. Clearly, if there is a well-developed domain theory so that the problem can be analytically solved by calculus, it is more convenient to pursue this approach. However, for many practical domains and applications it is difficult to construct such theories.

Unlike traditional systems, such as calculus-based models or enumerative schemes, GAs derive their power from the fact that they are capable of using experiences gained in previous exploitation and exploration for future search strategies. Since genetic algorithms work with a population of state vectors s , instead of a single state vector, they do this very efficiently in time. Within the theoretical framework of the GA this is sometimes referred to as implicit parallelism [16]. The complexity of the design of GAs does not increase with the number of individuals in a population or size and consistency of the domain. Only the number of calculations will increase. Although, unlike traditional systems, GAs often perform relatively better in large and complex

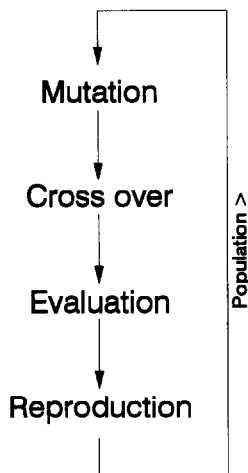


Fig. 2. A population of state vectors is subjected to changes and evaluations in an iterative process.

domains, the basic execution cycle is simple (Fig. 2).

The elements in the execution cycle will be discussed in the following section. The reader who is not interested in a description of the genetic algorithm can continue with the Experimental Section.

The evaluation function

Performance measures for evaluating the score, usually called the fitness, of alternate physical structures of PET should be chosen carefully in order to reflect the idea of good or bad solutions. In general, the fitness function is composed of two functions [17]:

$$U(s) = g(f(s))$$

where $f(s)$ is the objective function and $g(s)$ transforms the value of the objective function into a measure of fitness. This function is applied to all state vectors s in the population in order to calculate the fitness of all trial solutions.

Reproduction

Reproduction is a process in which individual solutions are copied in proportion to their fitness. Solutions with high scores, in other words chro-

mosomes having high fitness values, have more chance of being reproduced, and are therefore stimulated in their efforts to reach an optimal solution. This operator is basically an artificial version of natural selection, i.e., it entails a Darwinian survival of the fittest among string creatures. To preserve a constant population size in each cycle, state vectors with low fitness values are excluded from the population. For reproduction a 'spinning roulette wheel' approach is used to select state vector parents [13]. The purpose of state vector parent selection in a genetic algorithm is to impart more reproductive chances, on the whole, to those population state vectors that are fittest. There are many ways to do this. A commonly used technique is roulette wheel state vector parent selection. The effect of this technique is to return a randomly selected parent. Although this selection procedure is random, each parent's chance of being selected is directly proportional to its fitness. On balance, for a large number of generations this algorithm will expel the least fit members and contribute to the spread of genetic material in the fittest population members.

Mutation and crossover

The genetic operators mutation and crossover provide changes in the population during the evolution process. The mutation operator $M(s_m)$ arbitrarily alters a number of genes in the population. Mutation prevents the genetic algorithm from ending in local optimal points in solution space. Contrary to Holland's bitstring representation of state vectors, in this application the state vectors are real-coded. As a result of this difference, the genetic operators are different from those originally proposed by Holland. When using bitstring representation, one type of mutation operator will suffice. By applying the bitstring mutation operator to a gene consisting of n bits, the steps between 2^0 and 2^n in solution space can be made. When real-coded values are used, in general two types of mutation operators are required. One is jump mutation, which alters gene values into another random value within the search domain, and adds exploring capacities to

the GA. The other is step mutation, which is responsible for small changes in genes, and thus governs the exploiting capacities. The influences of these operators on the GA performance is described in ref. 16.

Simple crossover $C(s_l, s_m)$ may proceed in two steps, in the way as described below. Imagine a state vector with j parameters. First, two state vectors are randomly selected from the population. Second, a breakpoint is selected: an integer position k along the state vector is selected, with a uniform probability between 0 and the number of genes in a chromosome. Two new state vectors are created by swapping all characters between the position k and j . When genes are independent, the average fitness of the two state vectors remains the same before and after crossover. Crossover governs information exchange between two state vectors and it is expected that the fitness of one of the reassembled state vectors is better than the fitness of the best original state vector. The improved state vector has a higher probability to create offspring. Thus, the action of crossover speculates on new ideas constructed from highly fit genes from past trials. Fig. 3 gives an example of the crossover and the mutation procedure.

EXPERIMENTAL

In a previous article [6] we presented the use of artificial neural networks (ANNs) in order to develop a model of the relation between physical structure and mechanical properties of poly(ethylene terephthalate) (PET) yarns. The relation

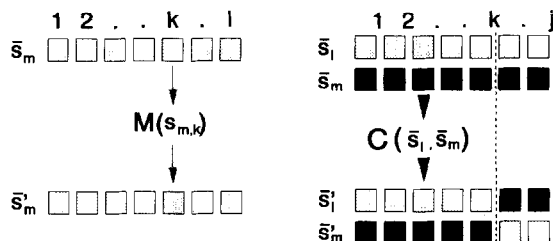


Fig. 3. The mutation operator randomly alters a randomly selected gene. The crossover operator recombines two randomly selected state vectors.

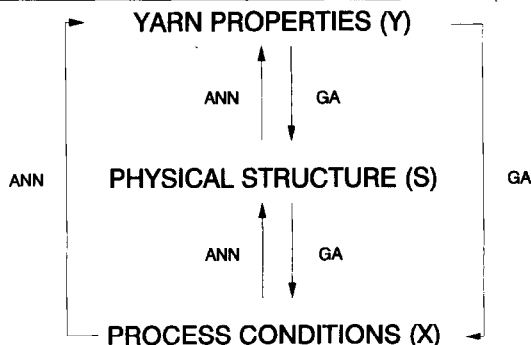


Fig. 4. The forward relations have been modelled by ANNs. The ambiguous reverse relation is optimized with GAs using the trained and validated ANN for the forward relation.

was studied on a set of 295 drawn yarn samples, resulting from an exceptionally large variety in applied process conditions. The ANN, consisting of three layers of neurons, was trained using physical structure and property measurements of these 295 yarns as input and output patterns respectively. All ten properties were fit to the physical structure within an acceptable variance for quantitative use.

In general, another interesting relation for process and product development is the reversed relation of the one discussed, viz. the relation of mechanical properties to physical structure and to process parameters. Development of yarns with optimum performance requires a detailed insight into the influences of process variations and molecular structure arrangements on the yarn properties. From an industrial point of view, it is therefore interesting to know which molecular structures are possible to achieve a combination of desired mechanical properties. Since this is an ambiguous and NP-complete relation, it is not possible to use only a neural network. Therefore an optimization technique is combined with an ANN trained for the forward relation in order to search for optimal solutions in the reversed relations (Fig. 4).

Since the number of (local) optimal points strongly depends on the combination of required properties, a robust search procedure is required. Genetic algorithms do not require any form of smoothness and are insensitive to the form and shape of the search space. The parameter setting

of the genetic algorithm is important for achieving evolutionary runs with high performances. The jump mutation, step mutation and crossover parameters are expressed as a probability of a gene or chromosome to undergo the operation. For instance, if the jump_mutation_probability is 0.05, each gene has a probability of 5% to change into another random value within the search domain. The chance of a chromosome to reproduce is directly related to its relative fitness.

A method for evaluating SYNGA

We let SYNGA predict the physical structure of PET in terms of the well-known two-phase model based on (thermo)mechanical properties. In order to evaluate the search behaviour of SYNGA we measured the five structure parameters and properties of eight PET yarns. The ability of the genetic algorithm to find the measured structure parameters depends on the number of global optimal points in solution space. The number of global optimal points is determined by: (1) the number of properties given to SYNGA for predicting the corresponding structure(s); (2) the uniqueness (of the combination) of values of the properties given to SYNGA for predicting the corresponding structure(s).

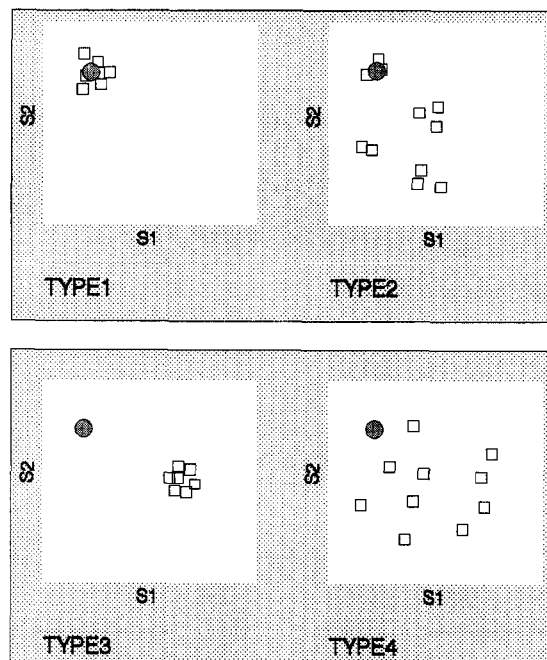
Testing of SYNGA is done under a variety of situations of 1 and 2. To discriminate between the search results of SYNGA, four types of search results relating to solutions that have reached a global optimum (i.e., that have reached the highest fitness value $U(s)$), can be distinguished.

Four types of search results

In search simulations in which a 'measured' result is known, four types of search results relating to the solutions that have reached the highest fitness values $U(s)$, can be distinguished.

Type 1. The 'measured' (or known) solution is found within experimental error and the other solutions are clustered around the 'measured' solution.

Type 2. The 'measured' solution is found and other solutions are found in other areas of the search space.



● measured solution

□ good solutions

Fig. 5. Four types of search results.

Type 3. The 'measured' solution is not found and the solutions are clustered within a small range from their centre of gravity. This phenomenon is known as premature convergence or as genetic drift [17].

Type 4. The 'measured' solution is not found and the population of solutions is clustered within a large range from their centre of gravity. This is known as a GA-hard problem [13].

These types of search results are shown in Fig. 5 for a situation in which two parameters are coded on the state vector s .

For classification into one of the four types, the Euclidian distances between all good solutions (solutions with high fitness values) and the 'measured' solution s ($D(s, s)$) were determined. So, for a good solution $s = [s_{1,1}, s_{1,2}, s_{1,3}, s_{1,4}, s_{1,5}]$ and a 'measured' solution $s = [s_{2,1}, s_{2,2}, s_{2,3}, s_{2,4}, s_{2,5}]$, $D(s, s)$ is calculated as

$$D(s, s) = \sqrt{\sum_{k=1}^5 (s_{1,i} - s_{2,i})^2}$$

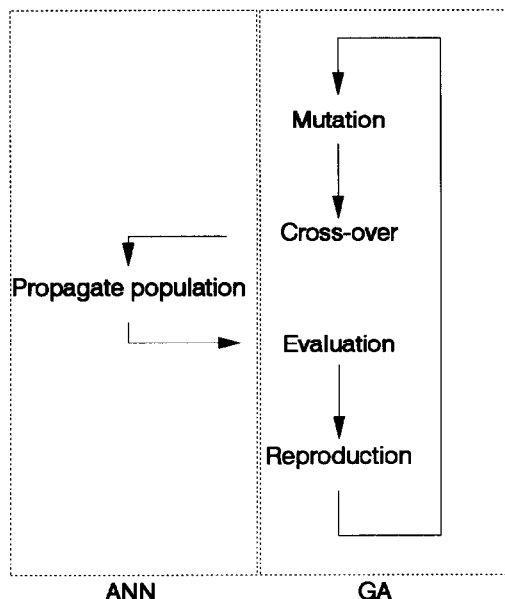


Fig. 6. Modification of the basic execution cycle of GAs. State vectors are propagated through the ANN and evaluated.

As a measure of concentration of solutions in the search space, the average Euclidean distance between the centre of gravity and the solutions (D) was calculated for each run.

When good solutions are found, it is possible to classify an evolutionary run into one of the four types of search results, using $D(s, s)$ and (D).

SYNGA's evaluation function

In this application s is a set of five yarn structure characterization measurements and f is a trained ANN which predicts the mechanical and shrinkage properties (Y_n) resulting on basis of s . So, in comparison with the basic execution cycle of genetic algorithms (Fig. 2), SYNGA's evaluation function is composed of two functions (Fig. 6). The first function propagates all state vectors through an ANN which has been trained for predicting PET yarn properties. The second function g transforms $f(s)$ into a fitness value.

As mentioned earlier, it has been recognized in GA theory that the evaluation function is vital, especially in multi-criteria decision making. In our choice of the evaluation function, the following prior conditions were taken into account: (1) The fitness of a state vector is equal to 1 only when all criteria are met, i.e. if the difference between $(f_n(s) - Y_n) = 0$ for all n . (2) A Gaussian distribution around $(f_n(s) - Y_n) = 0$, with a vari-

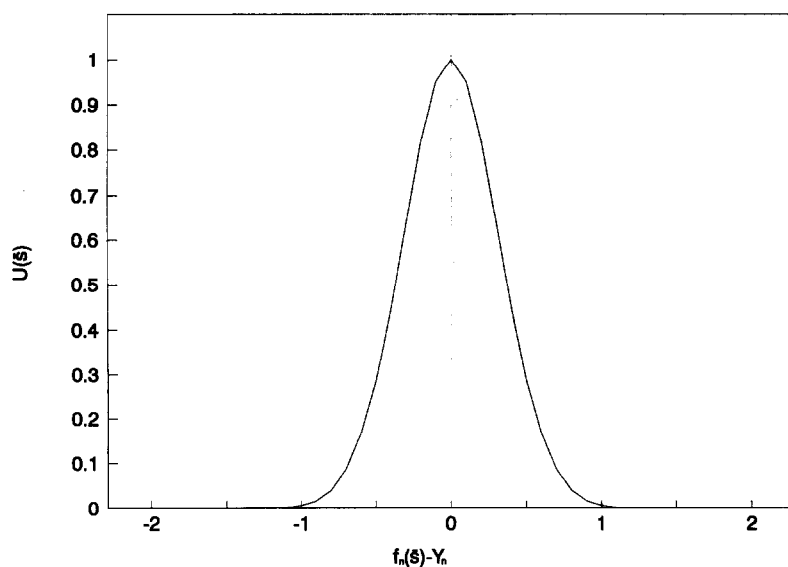


Fig. 7. Shape of $U(s)$ for $i = 1$.

TABLE 1

Parameter setting for this GA

Jump mutation probability	0.06
Step mutation probability	0.08
Crossover probability	0.06
Population size	60
Reproduction probability	0.07

ance of the same order of magnitude as the experimental error in the properties.

For g , therefore, the following mathematical formulation has been chosen:

$$U(s) = \left(\prod_{n=1}^i \exp \left(\frac{-[f_n(s) - Y_n]^2}{2\sigma} \right) \right)^{1/i}$$

where i is the number of yarn properties involved and Y_n is the desired value of property n . Y_n , $f_n(s)$ and s are scaled to attain equal treatment. For $i = 1$, $U(f_n(s) - Y_n)$ is shown in Fig. 7.

Simulation results

The problem with setting parameters for the GA is that an optimal parameter setting is highly dependent on the complexity of the question. In other words, the optimal parameter setting is problem specific. Since, as far as time is considered, it is not practical to tune the parameter setting to each problem, we optimized the parameter setting by performing a factorial design on a problem with average complexity. Jump and step mutation are related to a probability that a gene in the population will be exposed to mutation. Crossover is related to a probability that a chro-

mosome is allowed to crossover. The GA parameters used in the simulations are listed in Table 1.

Two data sets were composed. Set 1 consisted of four PET yarns with regular mechanical and shrinkage properties. Set 2 consisted of four PET yarns with highly specific properties. From all these yarns, five physical structure variables were measured as described in ref. 7. So, from all yarns physical structures as well as mechanical properties were known. The measured physical structure is denoted by s , and the measured property is denoted by Y .

To evaluate the performance of the search behaviour of SYNGA, the following experiments were carried out. First, of each yarn, the value of one property, its tenacity at rupture (Ten-Ru) was offered to SYNGA to recapture a set of possible physical structures corresponding to this property. To make the problem more specific, runs were carried out during which two, three, four and ultimately five properties of each yarn were offered to SYNGA. The properties were chosen as independently as possible. The setup of the experiments is depicted in Fig. 8. All experiments were carried out twice.

In total, 80 GA runs were carried out. In the following discussion only high-fitness solutions (for which $U(s) > 0.9$) are taken into account. Since all structure parameters are scaled between -1 and 1 the maximum distance in solution space is $\sqrt{(5 * 2^2)} = \sqrt{20}$. From laboratory measurements the critical 2σ $D(s, s)$ value is determined to be 0.19.

In Table 2, the simulation results of set 1, the set consisting of PET yarns without exceptional

TABLE 2

Number of GA runs ending in Type 1/Type 2/Type 3 or Type 4 results dependent on the number of criteria to be optimized of the set with regular mechanical and shrinkage properties

Number of criteria	Result				Number of solutions
	Type 1	Type 2	Type 3	Type 4	
1	1	0	1	6	113
2	1	1	2	4	83
3	3	0	1	4	88
4	2	3	3	0	68
5	4	2	1	1	66

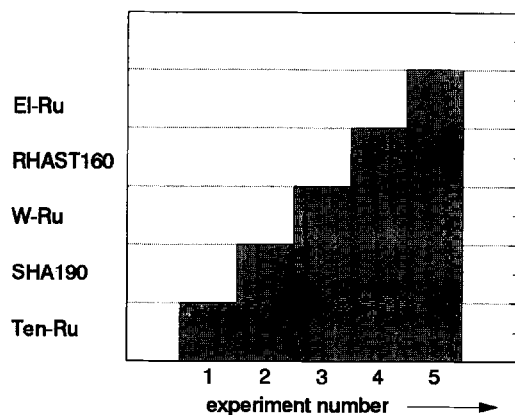


Fig. 8. Evolutionary runs for predicting physical structures of PET yarns given the following properties. For each yarn five experiments were carried out twice. With eight yarns this resulted in 80 GA runs.

properties, are shown. The final populations are evaluated as described in the section '*Four types of search results*'. In all 80 GA runs, solutions with fitnesses over 0.9 were found. A typical performance plot is shown in Fig 9.

The number of molecular structures found decreases with the number of corresponding properties involved or, in other words, as the number of criteria increases, the number of solutions decreases. Also note that the type of solutions gradually changes from Type 4/Type 3 to Type 2/Type 1. However, even if a combination of five properties is asked, there are still simulations which result in primary convergence or appear to be GA-hard. This indicates that the population size is too small. Simulations in which use has been made of a population size of 120 state

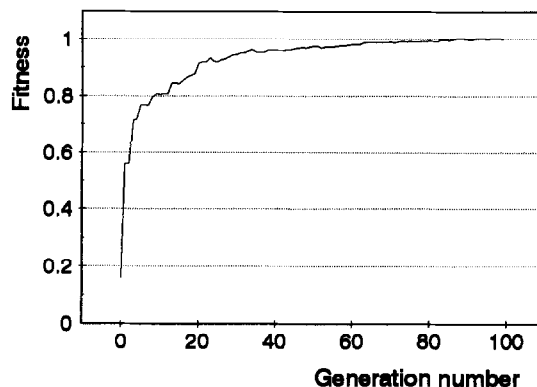


Fig. 9. Fitness of the best physical structure in the population. Notice that small improvements as well as large improvements occur as a result of successful step and jump mutations.

vectors indicate that, in comparison with a population size of 60 state vectors, there is a tendency towards less prematurely converged solutions. The results of data set 2 are shown in Table 3. Note that many Type 1 and Type 2 results are found when two properties are offered.

The structures that were found in most cases led to Type 1 results, and in two cases to Type 2 results. From this finding, we may assume that there are only a few optimal points in solution space and that the jump mutation operator, responsible for exploration, works properly.

A PRACTICAL EXAMPLE

In Fig. 10 a so-called microfibril has been depicted. A microfibril is a small structural unit with a diameter of about 10 nm, consisting of

TABLE 3

Number of GA runs ending in Type 1/Type 2/Type 3 or Type 4 results depending on the number of criteria to optimize, applied to a data set with extreme properties

Number of criteria	Result				Number of solutions
	Type 1	Type 2	Type 3	Type 4	
1	2	0	2	4	94
2	5	0	0	3	88
3	6	1	1	0	74
4	6	0	2	0	70
5	6	2	0	0	66

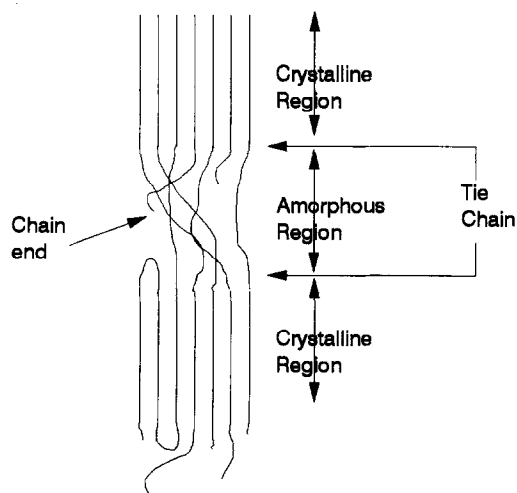


Fig. 10. Microfibril of PET consisting of alternating crystalline (ordered) and amorphous (less ordered) regions.

alternating crystalline (ordered) and amorphous (less ordered) regions. The cross-section of a usual PET filament contains millions of microfibrils which are very well aligned parallel to the fibre axis.

Most of the mechanical properties are strongly determined by the molecular arrangement within the amorphous regions. The parameters used for the description of the physical structure are [6,7]: $1 - V_c$: the volume fraction of amorphous material; η : the yarn viscosity as a measure for molecular length; Fas: the amorphous orientation fac-

tor; Fas/Fab: the contour length distribution of the tie chains between successive crystals; G : the average volume of the individual amorphous regions.

An ANN has been trained on the relation between the physical structure and the mechanical properties. In Table 4 a set of structural parameters is given together with some mechanical properties as calculated by the ANN. The physical structure and related properties are representative for a yarn with high strength and high thermostability. Suppose that the target is to improve the tenacity (Ten-Ru) from 664 to 750, combined with a reduction of the shrinkage (SHA190) from 6.6 to 5.0. These improvements in strength and thermostability have to be realized without substantial changes in the elongation at rupture (El-Ru) and residual shrinkage force (RHAST160). This example should illustrate that, in practice, always a combination of mechanical properties is required for any application.

In Table 5 the effect of the various structure parameters on the mechanical properties involved, as obtained from the trained ANN, are indicated [6]. A + indicates that an increase of the structure parameter causes an increase of the corresponding mechanical property.

From the table it is clear that a decrease of $1 - V_c$ is in favour of both a decrease of the shrinkage and an increase of the tenacity. The requirement of constant elongation at break and

TABLE 4

Physical structure and neural network property prediction of a virtual PET yarn. The targets for possible product improvements are given in column 2. In column 3, 4 and 5 the best solutions found with three successive GA runs are shown

	Initial product (ANN)	Target	Improved product		
			GA1	GA2	GA3
$1 - V_c$	0.360		0.395	0.393	0.393
η	1.700		1.840	1.835	1.849
Fas	0.800		0.832	0.829	0.829
Fas/Fab	1.300		1.372	1.382	1.360
G	7.0		11.4	10.8	11.2
SHA190	6.63	5.0	4.99	5.26	5.24
RHAST160	34.2	34	33.3	33.0	33.2
Ten-Ru	664	750	749	750	750
El-Ru	9.9	9.9	9.7	9.7	9.7

TABLE 5

Qualitative main effects of the five structure parameters on the properties considered in this article

	$1 - V_c$	η	Fas	Fas/Fab	G
SHA190	+	0	+	–	–
Ten-Ru	–	+	+	–	–
El-Br	+	0	– –	+	0
RHAST160	+	–	++	0	0

constant residual shrinkage force will temper the decrease of $1 - V_c$ somewhat. It is also clear that the tenacity can considerably be improved by using a higher viscosity without affecting the other parameters too much. Only the RHAST160 will slightly temper the effect. Both structural effects mentioned can be understood physically very well [7] but further explanation is beyond the scope of this paper.

With respect to the effect of the three remaining structural parameters, it can be seen from the table that changes which are favourable for an increase in tenacity also cause an increase in shrinkage in conflict with the requirements. So, it may be illustrated that, in practice, on a physical basis at best for some structural parameters a direction of the changes can be indicated, viz. when the changes result only in improvements of all mechanical properties involved. For most structural parameters however, the effect of changes differ from favourable to bad for the various properties. So, the application of a GA is very essential to show the way for effective structural changes for realizing a new set of mechanical properties as required. The practical situations are by far too complicate to oversee, from a physical point of view, all consequences of structural changes, and so a quantitative forecast of the physical structure, that is best suited for a given set of property requirements, is impossible at the moment. The GA, to the contrary, has a quantitative overview by means of the trained ANNs and is capable to indicate the structural adaptations for meeting new mechanical requirements. In Table 4 the structure parameters put forward by the GA to meet the requirements of

improved tenacity in combination with a shrinkage decrease at constant elongation at rupture and residual shrinkage force are given. In the three columns the best situation of each of three GA calculations has been presented. From these results it can be seen that a decrease of the amount of amorphous material ($1 - V_c$) and an increase of the viscosity η , as expected from physical considerations, are used by the GA indeed. The eventual structures as found by applying the GA three times are nearly identical and the desired properties could be approached very closely.

CONCLUSIONS

We have shown that genetic algorithms are capable of finding optimal molecular structures related to a variety of (thermo)mechanical properties of PET yarns, when use is made of a trained and validated ANN. Now, SYNGA is frequently consulted to predict the molecular arrangements of new desired types of PET yarns. Knowing this, it is easier to develop adapted or new processes to produce new yarn types.

Hybrids of artificial neural networks and genetic algorithms in the way described in this paper have a higher existensibility than this particular application alone. Since ANN are now widely used in, for instance, multivariate quantitative structure–activity relationships (QSARs) and process simulations, inversion of trained and validated neural networks becomes attractive. Multi-criteria decision making leads in general to a solution space with more (local) optimal solutions than single criteria optimizations. It becomes inevitable to use optimization techniques, such as GAs, that are capable of exploring a large and complex search space in an intelligent way in order to find values close to the global optimum.

ACKNOWLEDGEMENTS

The authors wish to acknowledge Akzo Research Laboratories Arnhem for their indispensable contributions and support to this work.

REFERENCES

- 1 D. Rumelhart, G. Hinton and R. Williams, Learning representations by back-propagating errors, *Nature*, 323 (1986) 533–536.
- 2 J.R. Long, V.G. Gregoriou and P.J. Gemperline, Spectroscopic calibration and quantitation using artificial neural networks, *Analytical Chemistry*, 62 (1990) 1791–1797.
- 3 T. Aoyama, Y. Suzuki and H. Ichikawa, Neural networks applied to quantitative structure–activity relationship analysis, *Journal of Medicinal Chemistry*, 33 (1990) 2583–2590.
- 4 J. Zupan and J. Gasteiger, Neural networks: A new method for solving chemical problems or just a passing phase?, *Analytica Chimica Acta*, 248 (1991) 1–30 (Review).
- 5 J.R.M. Smits, L.W. Breedveld, M.W.J. Derksen and G. Kateman, Pattern classification with artificial neural networks: classification of algae, based upon flow cytometer data, *Analytica Chimica Acta*, 258 (1992) 11–25.
- 6 A.P. de Weijer, L. Buydens, G. Kateman and H.M. Heuvel, Neural networks, used as a soft-modelling technique for quantitative description of the relation between physical structure and mechanical properties of poly(ethylene terephthalate) yarns, *Chemometrics and Intelligent Laboratory Systems*, 16 (1992) 77–86.
- 7 H.M. Heuvel, L.J. Lucas, C.J.M. van den Heuvel and A.P. de Weijer, Experimental relations between physical structure and mechanical properties of a huge number of drawn poly(ethylene terephthalate) yarns, *Journal of Applied Polymer Science*, 45 (1992) 1649–1660.
- 8 M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.
- 9 G.F. Miller, P.M. Todd and S.U. Hegde, Designing neural networks using genetic algorithms, in J.D. Schaffer (Editor), *Proceedings of the Third International Conference on Genetic Algorithms*, George Mason University, Fairfax, VA, 1990, pp. 379–384.
- 10 D.J. Montana and L. Davis, Training feedforward neural networks using genetic algorithms, *Machine Learning*, (1990) 762–767.
- 11 M. Bos and H.T. Weber, Comparison of the training of neural networks for quantitative X-ray fluorescence spectrometry by a genetic algorithm and backward error propagation, *Analytica Chimica Acta*, 247 (1991) 97–105.
- 12 L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.
- 13 D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA, 1989.
- 14 C.B. Lucasius and G. Kateman, Application of genetic algorithms for optimization problems in chemometrics, *Trends in Analytical Chemistry*, 10 (1991) 254–261.
- 15 J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.
- 16 J.J. Greffentette and J.E. Baker, How genetic algorithms work: A critical look at implicit parallelism, in *Proceedings of the Third Conference on Genetic Algorithms*, pp. 20–27.
- 17 H.A. de Jong, *Analysis of the Behaviour of a Class of Genetic Adaptive Systems* (Doctoral Dissertation), Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI, 1975.